**How to Use BitBucket’s Version Control**

Carolyn Anderson

*last updated May 20, 2015*

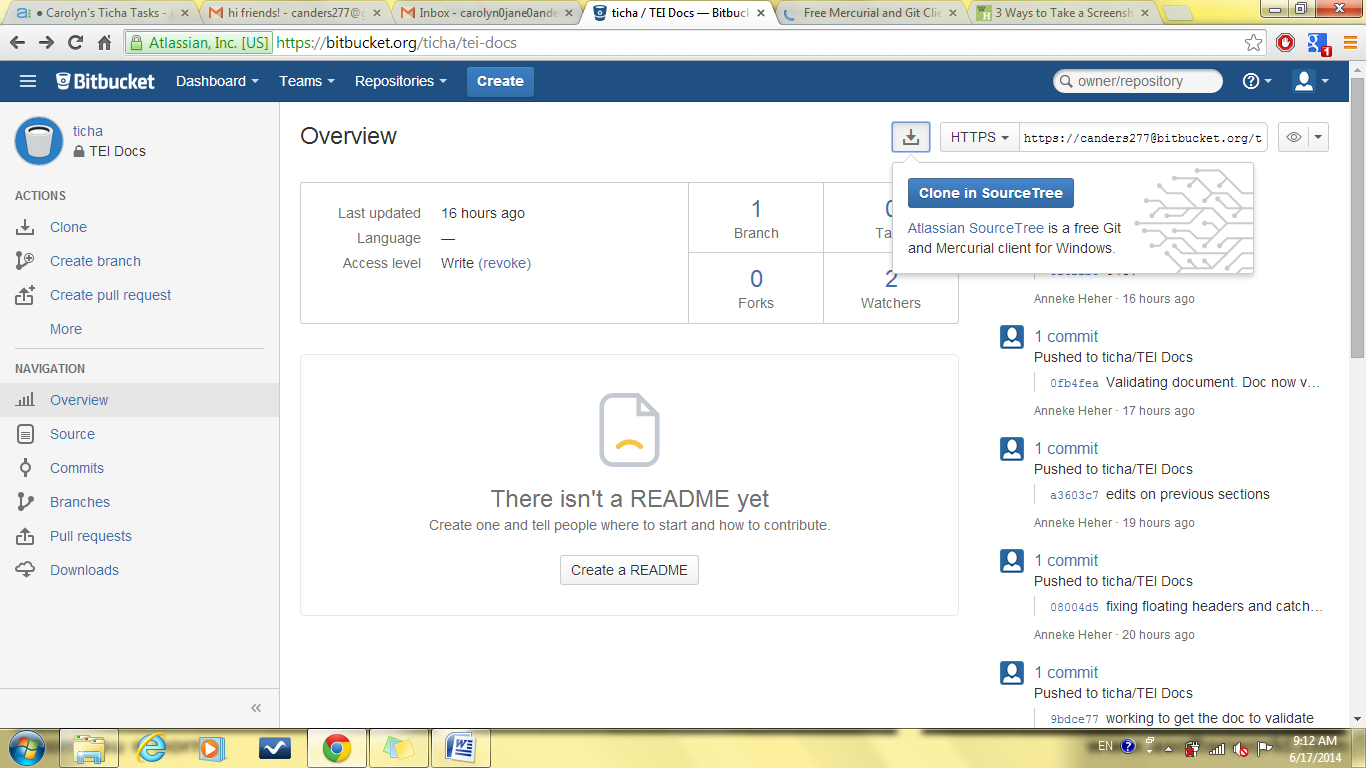
Version control allows multiple people to edit a document at the same time. The Learn Mi’gmaq project uses BitBucket to keep track of all files that the website relies on. This allows us to go back to a previous version if we accidentally modify a file.

The software at the heart of the system is GIT, which is used by many other websites as well. In a Git system, all files are stored in a central **repository** (repo), usually online. We will be using the site BitBucket to host our repository.

One way to access a repository is using the command line on your computer. However, Bitbucket is compatible with a program called **Atlassian Sourcetree** that provides a more visual interface. The rest of the tutorial assumes that you will be using Sourcetree, but the general framework holds true for command line use as well.

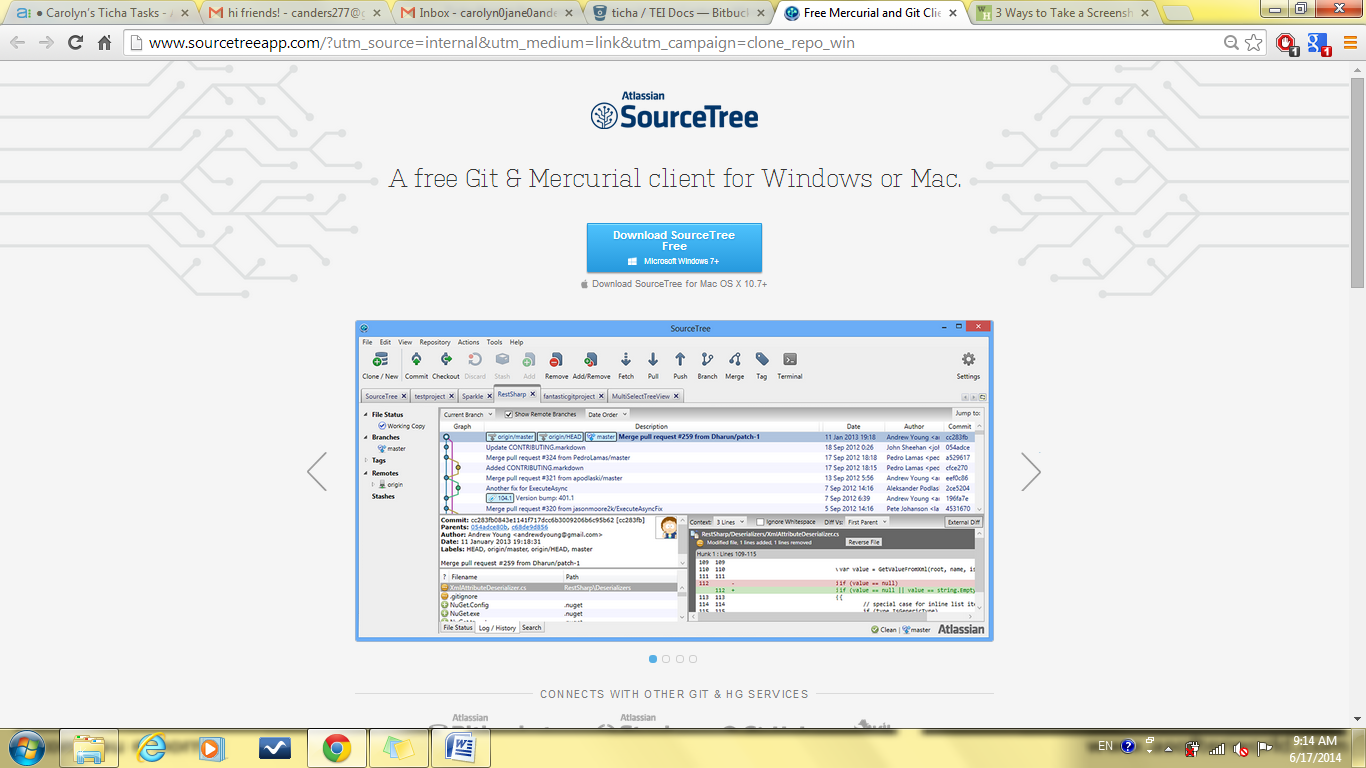
**Getting Started**

1. In order to use Bitbucket, you will need create a Bitbucket account. Once you have done this, you will be invited to join the Learn Mi’gmaq repository by someone else on the project. It is called canders277.bitbucket.org.
2. Next, you will need to download Sourcetree. Go to the overview page for the Learn Mi’gmaq project on Bitbucket. On the right-hand side of the Overview title, there will be a download image. Clicking on it will give you an option of downloading Sourcetree.



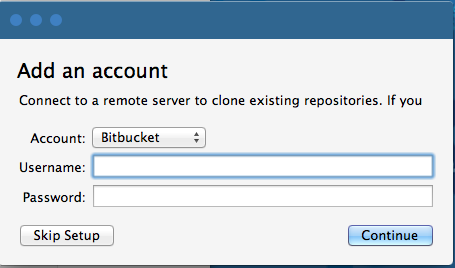
**Figure 1. Ticha Overview page with link to Atlassian SourceTree.**

1. You will be taken to the Sourcetree site. Download the version for your operating system.



**Figure 2. Atlassian Sourcetree download page.**

1. You may be prompted for your login information as part of the set-up. If so, enter it.



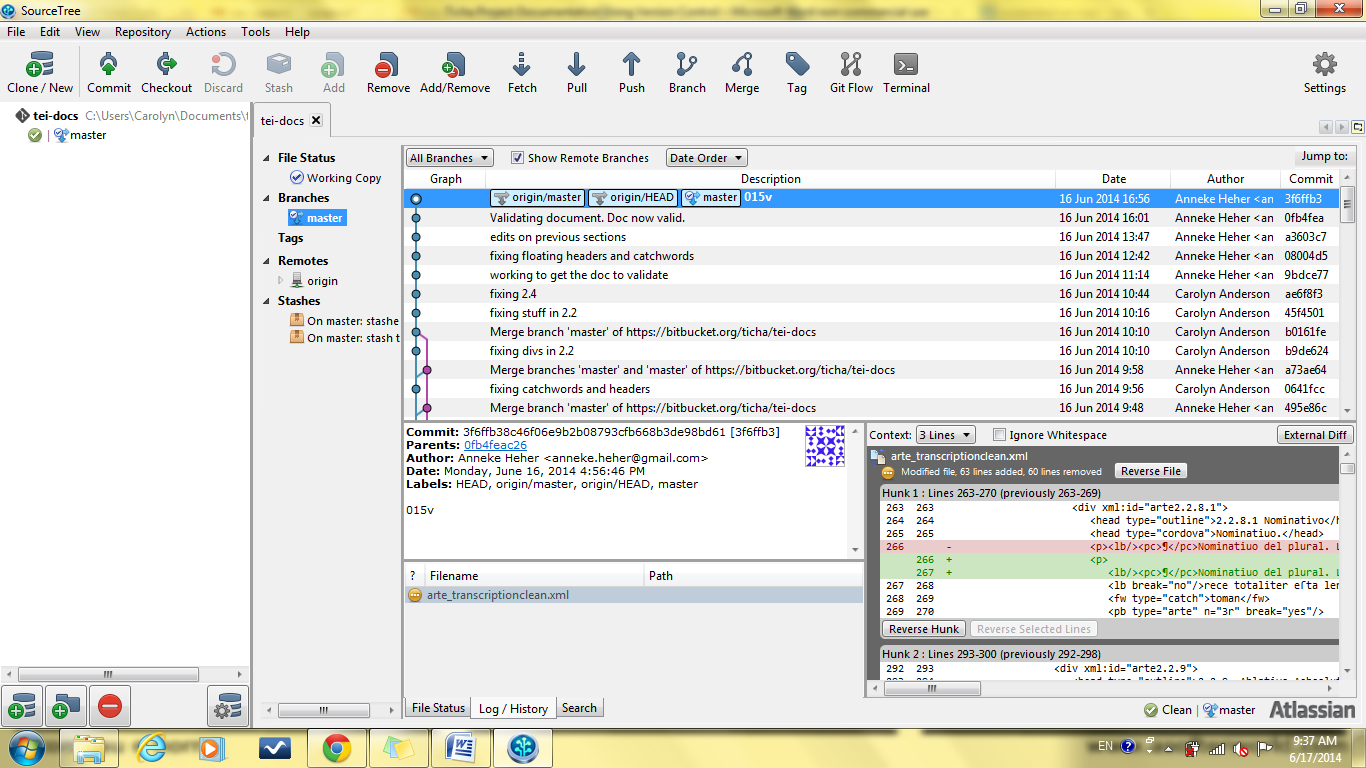
**Figure 3. Setting up Sourcetree.**

1. To work with the files, the first step is to **clone** the repository. This creates a copy of the repository on your computer. This is called a **local copy**. If you are prompted to do this during setup, select the **canders277/canders277.bitbucket.org repo**. You can also go back to the Bitbucket website and click on the same download icon. This time, choose the option **Clone in Sourcetree. Make sure to notice what destination folder the repo is placed in.**
2. Now you can open and edit the files on your own computer.

**Using version control**

A central idea in using Git is branching. When you clone a repository and make changes to it, your copy of the files **branches** off from the **master** (the central repository). Unless you want to end up with a different set of files for every person that edits the repo, at some point your branch will need to **merge** back into the master.

You can view the branching history of the repository by clicking on **master** under **Branches** in Sourcetree.

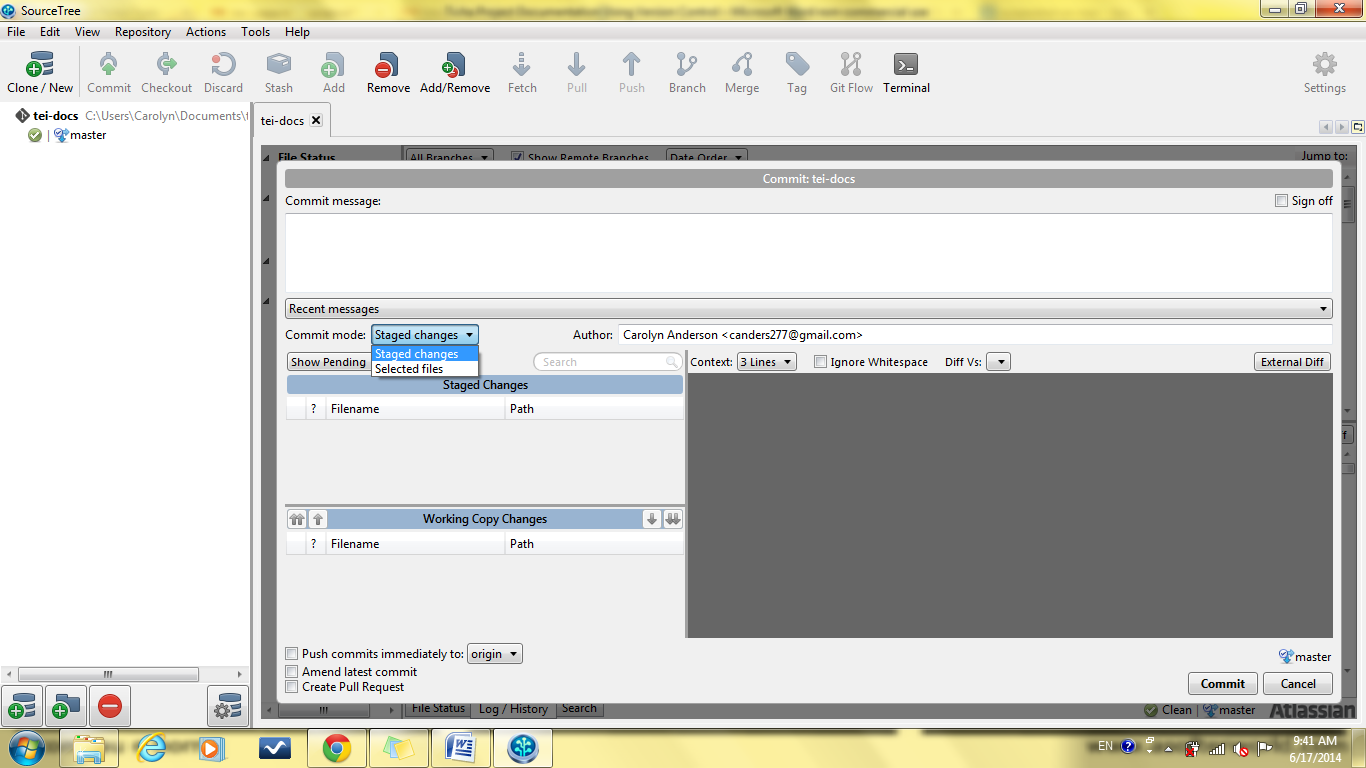


**Figure 4. Branching history.**

One way that your local copy may diverge from the master is when you change the files. In this case, you want to update the master branch with the changes you have made. There are several steps in this process.

1. To begin, click the commit button across the top of Sourcetree.
2. First, you need to tell the system which changes you would like to propagate to the master. This is called **staging**.

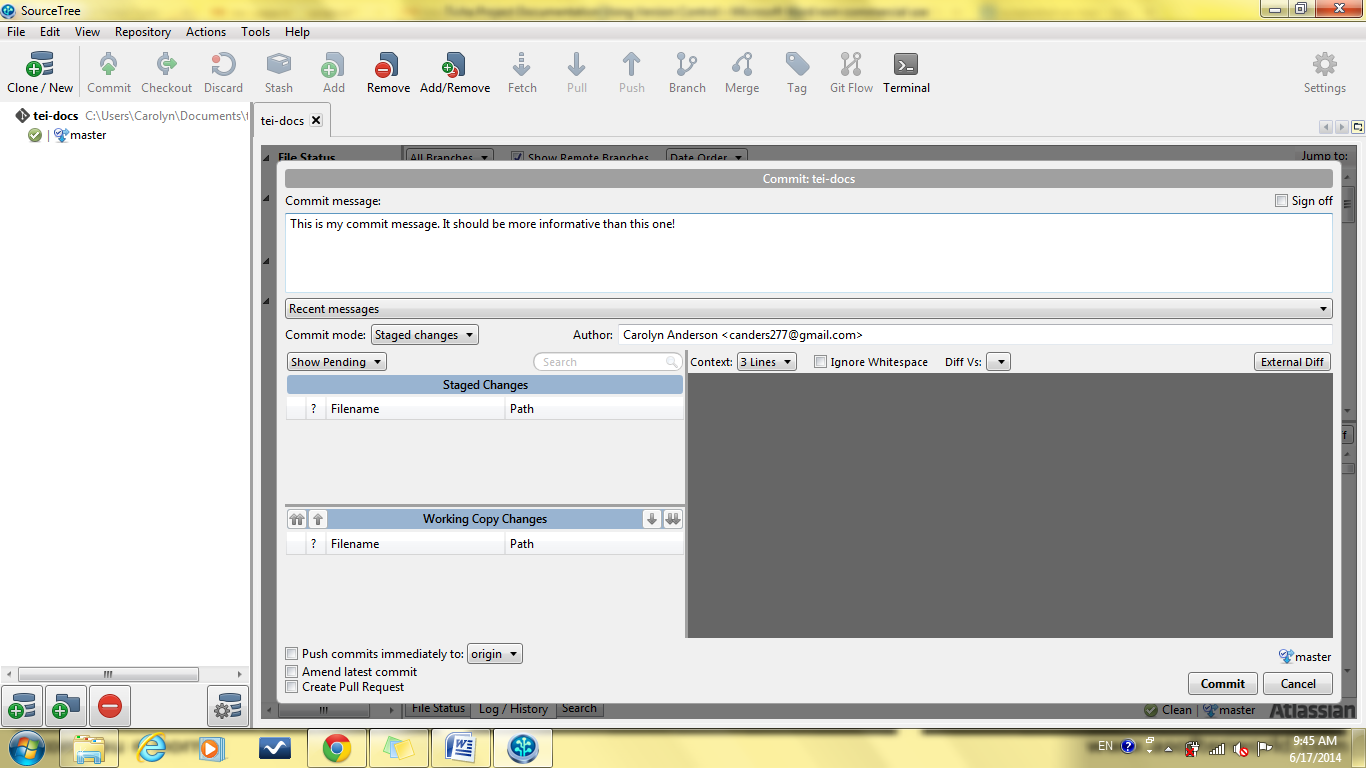
You can select entire files or chunks of files. You can stage all changes by changing the **commit mode** on the left, or select chunks to stage in the window on the right.



**Figure 5. Commit box with staging options.**

1. Next, you need to **commit** your changes. This bundles the changes you have made into one update and allows you to add a message describing your edits.

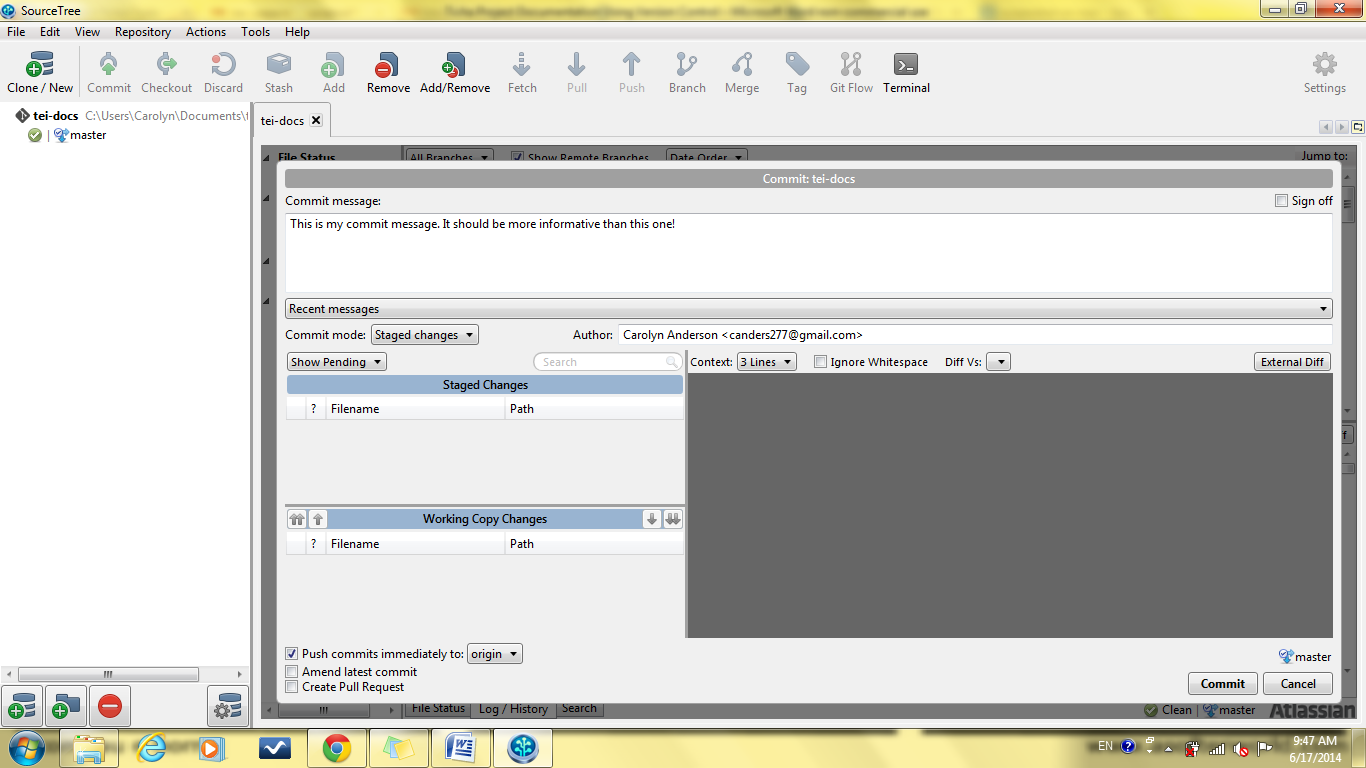
It is good practice to write descriptive commit messages in case you decide to revert your changes in the future.



**Figure 6. Commit box with commit message.**

1. Last, you need to **push** your changes. This sends the changes to the master to update it.

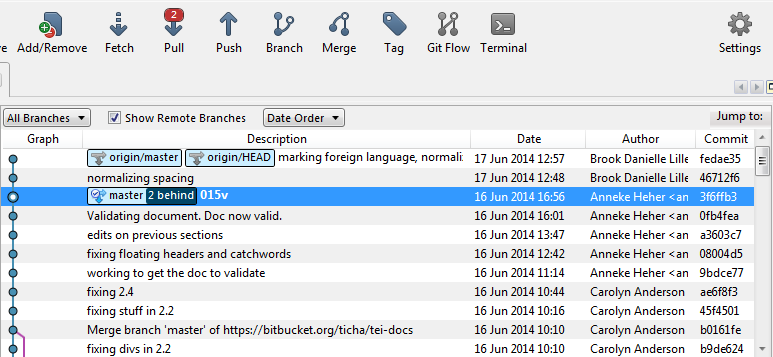
You can do this automatically when you commit by checking the box labeled **Push commits immediately to origin** in the left hand corner of the commit box, or you can do it by clicking on the push button across the top of Sourcetree.



**Figure 7. Pushing immediately when committing.**

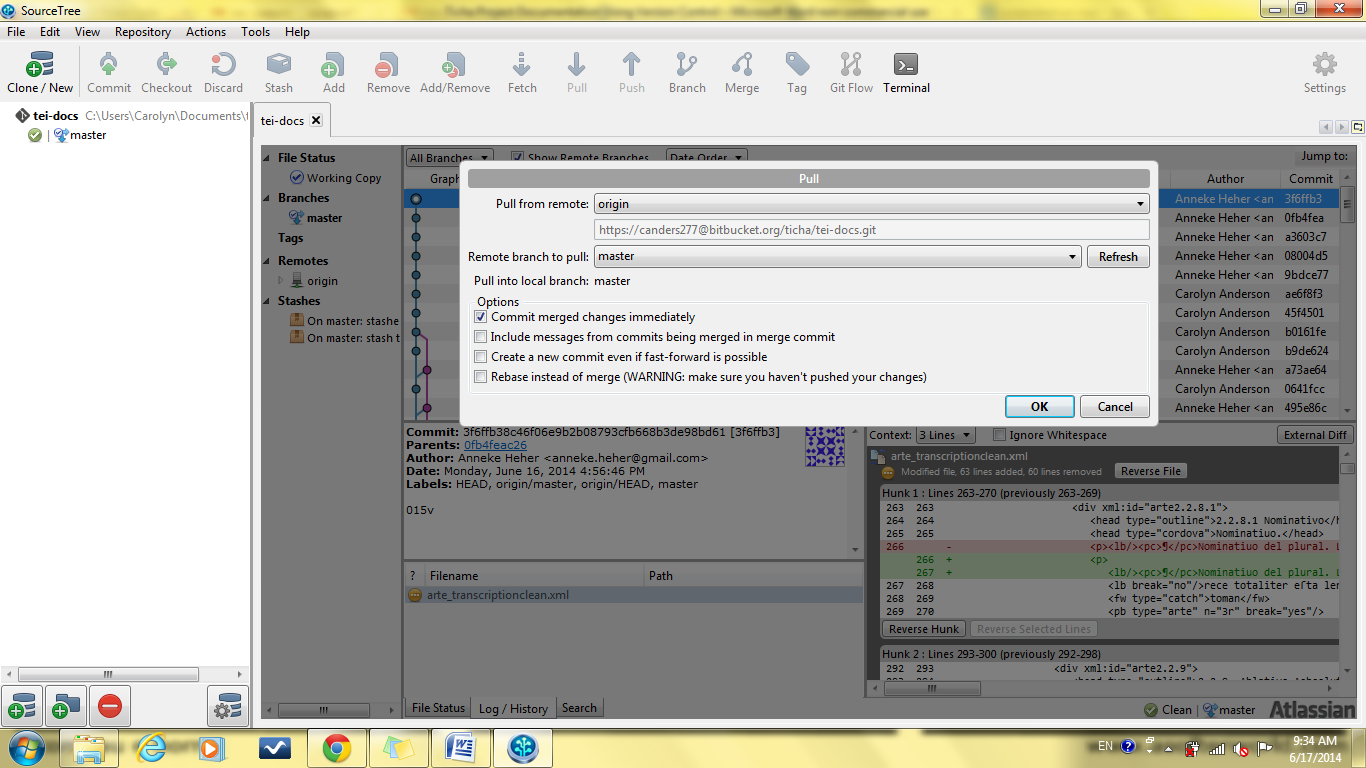
Another way your copy may diverge is when someone else makes changes and pushes them to the master. In order to catch up, you need to accept their changes. This is done by **pulling**. Pulling updates your local copy with changes that have been made to the master by other users.

1. Sourcetree will usually show you when you need to pull by displaying a red number on the pull button across the top.



**Figure 8. Pull needed.**

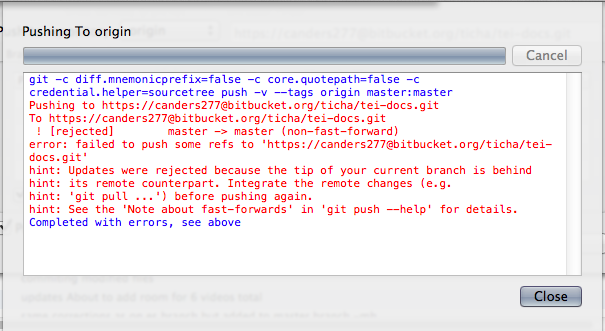
1. In order to pull, click this button. The following dialogue box will appear. Check **commit merged changes immediately** and press **OK**.



**Figure 9. Pulling updates.**

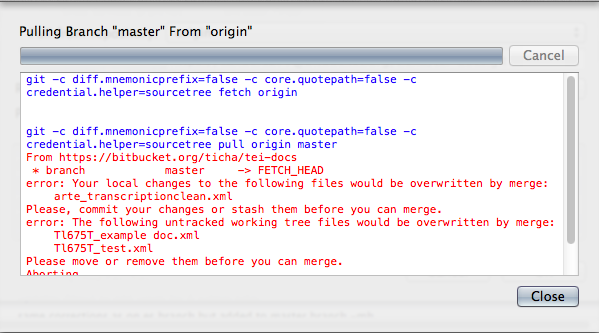
1. Afterwards, you should see a new commit message appear on the branch graph.

The order of these steps matters. First, Sourcetree will not allow you to push changes unless your own branch is up-to-date. This means you will need to pull any changes before you can push your own.



**Figure 10. Error: pushing from a branch not up-to-date.**

Second, if you pull before you have committed your own changes, your edits may be discarded. Sourcetree will give you a warning before this happens, but it is good to commit any changes before pulling.



**Figure 11. Warning: uncommitted changes.**

**Merge conflicts**

Most of the difficulty in using version control comes from **conflicts** between copies.

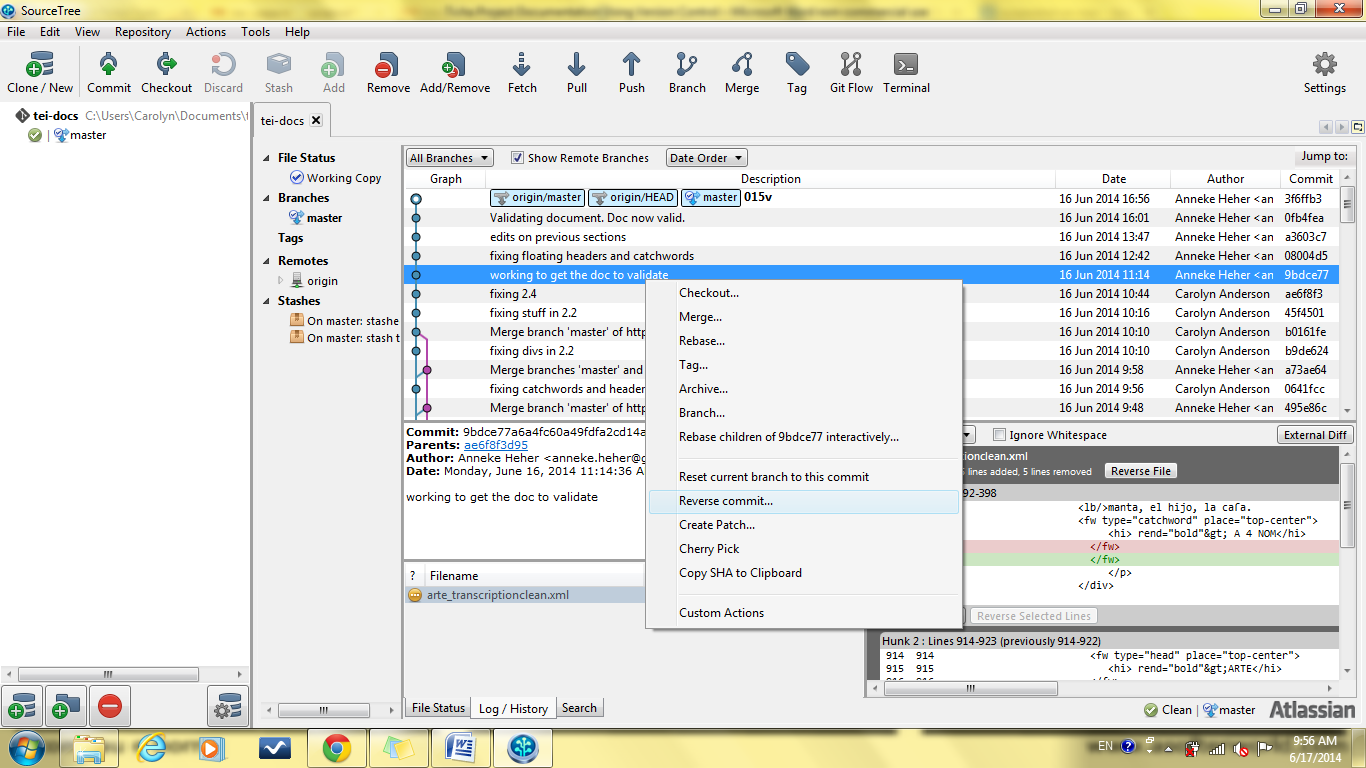
Conflicts arise when two people edit the same section. While version control software automatically merges the changes that simultaneous users make to the document, it is difficult to resolve changes made to the same section. If two people make changes to the same section, the software does not know which changes to keep. Often, it will take the first changes pushed.

If you have made changes to the same section that someone else has just pushed changes to, when you try to commit Sourcetree will give you a warning message in the commit message box that your copy conflicts with the master branch. This is a bad situation.

**Stashing changes**

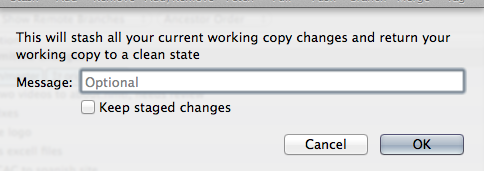
The best solution to this situation is to use stashing. **Stashing** allows you to save local changes in a separate location, merge with the master, and then reapply the stashed changes. If you have already committed your changes, you will need to roll back your commit:

1. Right-click on your commit and select Reverse commit.



**Figure 12. Reversing a commit.**

1. Next, click the stash button across the top of Sourcetree.
2. Stash your changes and name the stash. This is essentially hiding away the changes you have made, so that you can reapply them later.



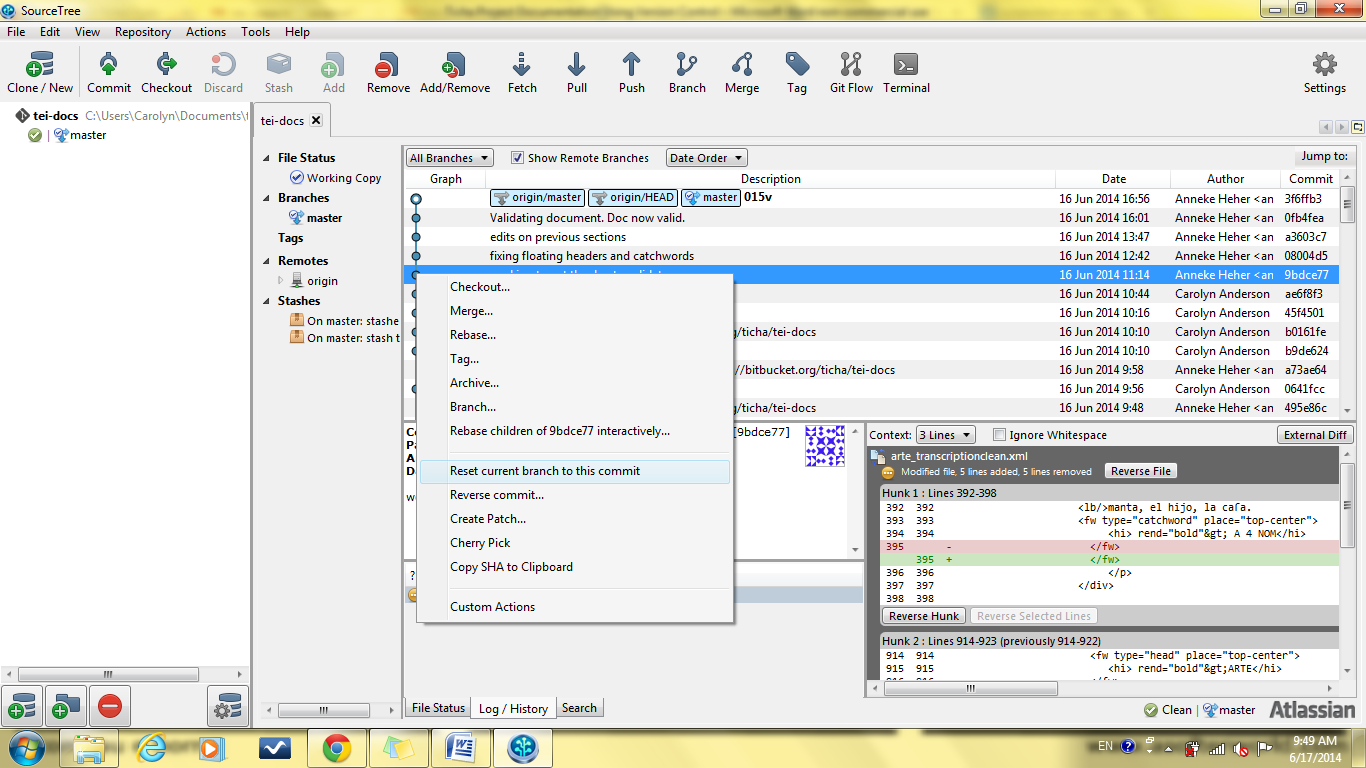
**Figure 13. Stashing changes.**

1. Now you should be able to pull.
2. After you are up-to-date, you should be able to reapply your changes from the stash.

**Discarding changes**

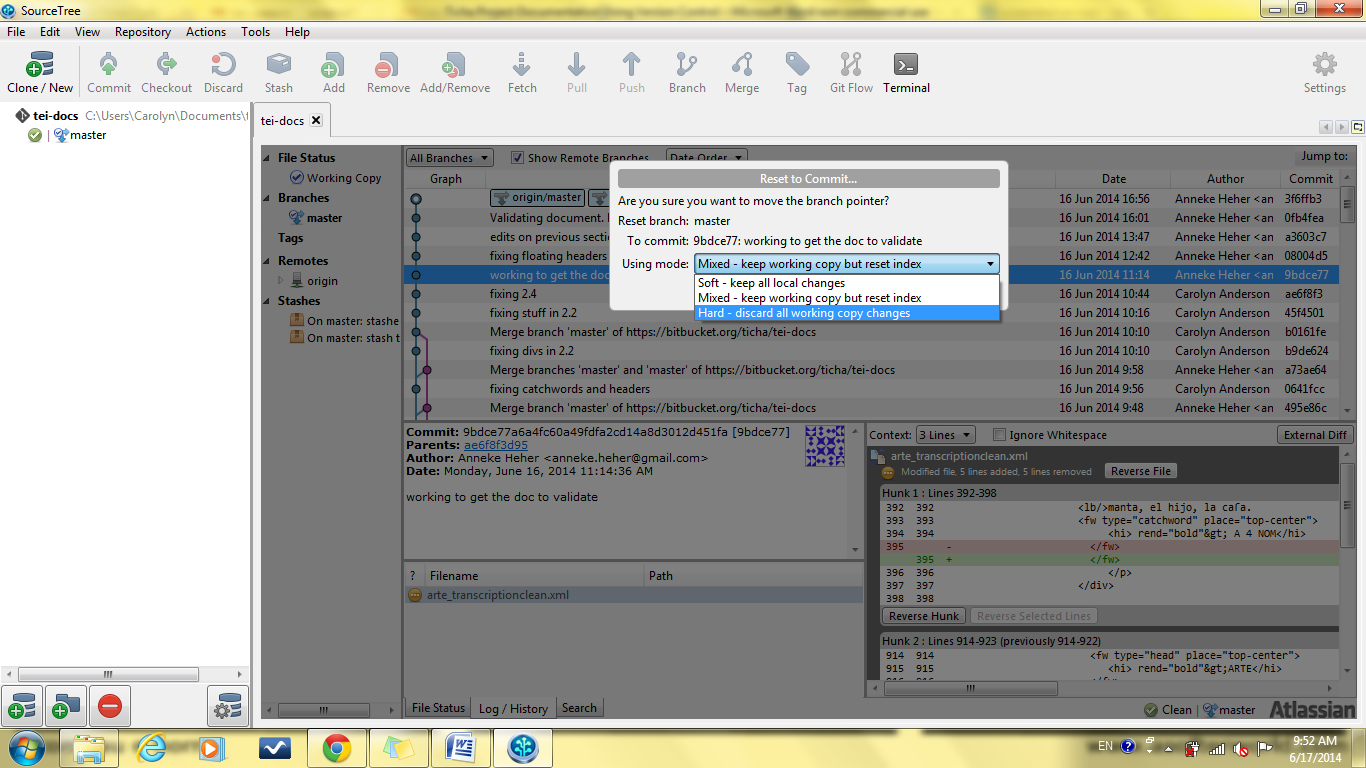
If stashing does not work, you may choose to discard your own changes:

1. Right-click on the most recent commit (the one that conflicts with your own changes) and select **Reset branch to this commit.**



**Figure 14. Fixing a conflict by resetting to previous commit.**

1. Select the option **Hard – discard all working copy changes.**



**Figure 15. Discarding working copy changes.**

Another solution is to have the other person roll back their changes. They should follow the instructions above, but click on the commit just prior to their conflicted commit.

Either solution results in lost time and work. It is much simpler to avoid this situation by communicating ahead of time about which section you are editing. ***Version control does not eliminate the need to communicate about where you are working.***